The MILC Benchmark

1. Introduction

Lattice QCD (LQCD) applications are parts of OLCF workloads. In a typical year there would be 2-3 large (and occasionally some small) LQCD projects running at OLCF. From a computational perspective LQCD applications tend to fall into two groups: gauge generation---which is a big hybrid molecular dynamics Monte Carlo process---and analysis---where the gauge configurations generated in the previous step are processed. Typically this involves the calculation of hadronic observables constructed from quark propagators on each configuration.

In general gauge-configuration generation tends to be a strong scaling problem, spread over as many nodes as is practicable. As the node count increases, the surface to volume ratio of the local problem decreases, and the problem becomes increasingly communications bound. Factors affecting the communication can include communications latency and bandwidth. Due to the recent decrease in the ratio of memory and network bandwidths, it is common for strong scaling to top out at O(100)-O(1000) nodes leading recently to a tendency to run multiple streams of gauge generation in a single job; essentially in the style of a small ensemble (e.g. 4-streams of 512 nodes or similar).

In contrast analysis makes use of the independence of the gauge configurations and is virtually almost run in an ensemble manner. The ensemble member sizes typically are chosen to have a sufficiently large local volume per node to keep strong scaling effects to a minimum. For hadronic observables, generally the lattice Dirac Equation needs to be solved which is stencil-like and is generally memory bandwidth bound. Analysis stages may also face limitations from I/O depending on whether they write out the solutions of the dirac matrix, or its eigenspectrum.

1.1 The MILC Benchmark

The MIMD Lattice Collaboration (MILC) code has been a workhorse of LQCD calculations utilizing Highly Improved Staggered (HiSQ) quarks. The code is written in C and can utilize libraries for performance improvement including the QUDA library for GPUs and the QPhiX library for CPUs. This benchmark is adapted from the NERSC-10 Latice QCD Workflow Benchmark (https://gitlab.com/NERSC/N10-benchmarks/lattice-qcd-workflow) which alternates the running of gauge generation using the `su3_rhmd_hisq` application and hadron spectroscopy calculations using the `ks_spectrum_hisq` application.

In this benchmark however, we care solely on the gauge generation part with `su3_rhmd_hisq` application. The `ks_spectrum_hisq` benchmark is not part of this benchmark.

2. Run Rules

The **OLCF-6 Benchmarks Run Rules** is to be followed. Offeror can enable and utilize existing performance optimization libraries such as QUDA on GPUs (and potentially QPhiX on AVX—capable CPUs) which are already integrated into the library (so no source code changes are needed). One note is that the QUDA library employs autotuning. Autotuning of a library is permitted. Although we are primarily interested in post-tunings timings, pre-tuning timings are useful to gauge tuning overhead.

3. Building MILC

The MILC code provided with this benchmark is revision `13ffa851` from the MILC GitHub repository (https://github.com/milc-qcd/). It was obtained with the following command:

```
git clone https://github.com/milc-qcd/milc_qcd.git milc_qcd
cd milc_qcd
git checkout 13ffa851
```

For using GPUs on GPU-based systems, MILC requires the QUDA library. The included QUDA was obtained using the following command:

```
git clone https://github.com/lattice/quda.git quda
cd quda
git checkout 35b57df9f
```

Instructions for building on Frontier is provided in the file BuildingOnFrontier.md (BuildingOnFrontier.md).

4. Benchmarks Details

4.1 Input Lattices

Similar to NERSC-10 benchmark, five concrete lattice sizes are provided as enumerated in the table below

Class (Name)	Lattice Size (lattice sites)	File Size
Tiny	48^3x64	1.9 GB (x2 files)
Small	64^3x96	6.8 GB (x2 files)
Medium	96^3x192	46 GB (x2 files)
Reference	144^3x288	231 GB (x2 files)

Target	192^3x384	730 GB (x1 file)

We use the same input lattice configurations as provided with NERSC-10 benchmark. Scripts to obtain the data and to validate results also come from NERSC-10 benchmark and have been provided here for reference.

4.2 Benchmark Requirements from Offerors

Offerors are required to provide benchmark times (or projections) for both the gauge generation and the runtime steps for the *target* problem size in the following table. Additionally optimized benchmarks may be reported.

	Ported node count	Ported Wall-clock time	Optimized Node count	Optimized Wall-clock time
Gauge Generation (su3_rhmd_hisq)				

We provide below results from Frontier for the "Reference" latice size. The "Total time" is as reported by the benchmark validation script as `Total time`. I/O time is the sum of the `Read time` and `Write time` as reported by validation script.

Generation Benchmark	**Node Count**	**Total Time (sec)**	**I/O time (sec)**	**Benchmark Time (sec)**
Frontier 'reference'	64	738.69	424.7	6300.30